

Doelstellingen

Tijdens dit labo maken we nog een oefening op DOM, zonder jQuery. De topics van vandaag zijn dus:

- DOM
- Events

Werkwijze, timing en upload

Dit labo loopt over één sessie en moet op het einde van de labosessie geupload worden via Toledo. Upload één enkele zip met daarin je oplossingen. De docent kan vragen de opdrachten thuis verder af te werken – je kunt dan wat extra punten verdienen. Neem die dan gewoon het volgende labo mee; je hoeft niet opnieuw te uploaden. Het huishoudelijk reglement is van toepassing.

Opgave – Binary Sudoku

Open de opgave voor de Sudoku. We zullen dit in verschillende stappen scripten tot een volwaardig Sudoku spel.

1.1 Verbinden van het script aan de HTML

Maak een script aan en verbind het aan je HTML; gebruik het standaard script encapsulation patroon in strict mode.

1.2 Variabelen

Maak drie variabelen aan voor volgende DOM node(s):

- de tabel zelf
- de rijen van de tabel
- alle cellen van de hele tabel

Vind hiervoor de juiste DOM selectoren; controleer met `console.log()`.

1.3 Cellen invullen

Denk na over wat moet gebeuren als een toets in een cel ingedrukt wordt:

- wat als het een 1 is? wat als het een 0 is?
- wat als het delete of backspace is?
- wat als het een willekeurig ander karakter is?

Behalve bij de delete of de backspace moet de cel ook focus verliezen (“blur”) telkens na het intikken van een karakter.

Welk event ga je gebruiken? Tip: (het is geen click event.

Je hoeft geen 36 event listeners te definiëren op alle cellen apart; je kunt ook gebruik maken van event bubbling en één enkele event listener definiëren op de tabel, zie [deze slide](#). Voeg dus een event listener voor dit event toe op de tabel. Inspecteer met `console.log()` het event object dat gegenereerd wordt. Bekijk grondig alle properties.

1.4 Aantal nog in te vullen cellen berekenen

Bereken het aantal nog in te vullen cellen, en pas daarmee het getal aan naast ‘Nog in te vullen:’ links onder. Maak gebruik van een zelf te schrijven functie. Roep deze functie op telkens een toets ingedrukt wordt, zodat het aantal automatisch aangepast wordt.

1.5 Knop ‘controleer’

Als op deze knop geklikt wordt, moet het bord gecontroleerd worden. De regels:

- a) het aantal enen en nullen op elke rij en kolom moet gelijk zijn
- b) een 1 of 0 mag niet meer dan 2 keer na elkaar voorkomen

1.5.1 Functie schrijven

Schrijf een functie die een array van cellen als parameter accepteert, en voorlopig enkel via `console.log()` de waarden van deze array weergeeft.

Roep nu deze functie op voor elke rij en kolom. Controleer of je de juiste waarden terugkrijgt.

Tip: voor de selectie van de cellen van een kolom kun je `table.querySelectorAll(...)` gebruiken in combinatie met een CSS3 pseudo class selector gebruiken.

1.5.2 Controle gelijk aantal enen en nullen

Controleer regel a). Voer deze controle enkel uit als alle cellen van een rij of kolom ingevuld zijn. Als het aantal enen en nullen niet gelijk is, geef je alle cellen een roze achtergrondkleur (er is in de CSS reeds een class `invalid` voorzien).

1.5.3 Controle op 2 of meer enen of nullen na elkaar

Controleer nu regel b). Dit onderdeel is iets moeilijker; misschien doe je best eerst onderdeel 1.6, en keer je daarna terug.

Geef elke reeks van 3 of meer gelijke cellen een roze achtergrondkleur.

1.6 Eindspel

Het spel is afgelopen als het aantal lege cellen 0 is en geen fouten voorkomen. Toon in dat geval een gepaste mededeling en disable het bord. Suggestie: maak alle cellen niet-editeerbaar, en maak het bord half transparant.

Tip: hoe kun je in één regel controleren of er geen enkele foutieve cel meer is?